## **IPL Dataset Project Report**

### **Introduction**

This project involves analysing the Indian Premier League (IPL) dataset using a MySQL database. We established relationships between various tables and wrote SQL queries to extract meaningful insights from the data.

### **Database Structure**

The database consists of several tables, including:

* ipl\_match: Contains details of each match.
* ipl\_player: Contains player information.
* ipl\_batsman\_scored: Records runs scored by batsmen.
* ipl\_player\_match: Links players to matches.
* ipl\_team: Contains team information.
* ipl\_venue: Contains venue details.
* ipl\_outcome: Contains match outcome details.

### **SQL Queries and Insights**

1. **Total Number of Matches Played**

SELECT COUNT(\*) FROM ipl\_match;

This query returns the total number of matches played in the IPL.

1. **First and Last Match Dates**

SELECT MIN(Match\_Date) AS first\_Date, MAX(Match\_Date) AS Last\_Date FROM ipl\_match;

This query provides the dates of the first and last matches played in the IPL.

1. **Total Number of Players**

SELECT COUNT(\*) FROM ipl\_player;

This query returns the total number of players who have participated in the IPL.

1. **Number of Fours Hit by Players**

CREATE TABLE fours AS

SELECT \* FROM (

SELECT pm.match\_id, p.player\_name, COUNT(runs\_scored) AS total\_four

FROM ipl\_batsman\_scored bs

JOIN ipl\_player\_match pm ON pm.match\_id = bs.match\_id

JOIN ipl\_player p ON p.Player\_id = pm.Player\_id

WHERE Runs\_scored = 4

GROUP BY pm.match\_id, p.player\_name

) AS A

ORDER BY total\_four DESC;

This query creates a table listing the number of fours hit by each player.

**5.Highest Win Percentage for Teams**

CREATE TABLE HighestPercentageofTeam AS

WITH Team\_Matches AS (

SELECT Team\_Id, COUNT(Match\_Id) AS Matches\_Played

FROM (

SELECT Team\_1 AS Team\_Id, Match\_Id FROM ipl\_match

UNION ALL

SELECT Team\_2 AS Team\_Id, Match\_Id FROM ipl\_match

) AS All\_Teams

GROUP BY Team\_Id

),

Team\_Wins AS (

SELECT Match\_Winner AS Team\_Id, COUNT(Match\_Id) AS Matches\_Won

FROM ipl\_match

WHERE Match\_Winner IS NOT NULL

GROUP BY Match\_Winner

)

SELECT t.Team\_Name, tm.Matches\_Played, tw.Matches\_Won,

(tw.Matches\_Won \* 100.0 / tm.Matches\_Played) AS Win\_Percentage

FROM ipl\_team t

JOIN Team\_Matches tm ON t.Team\_Id = tm.Team\_Id

LEFT JOIN Team\_Wins tw ON t.Team\_Id = tw.Team\_Id

ORDER BY Win\_Percentage DESC

LIMIT 1;

This query calculates the highest win percentage for teams.

**6.Average Score of Each Team in the Season**

CREATE TABLE AverageScoreofTeam AS

WITH Team\_Scores AS (

SELECT m.Season\_Id,

CASE

WHEN bs.Innings\_No = 1 THEN m.Team\_1

WHEN bs.Innings\_No = 2 THEN m.Team\_2

END AS Team\_Id,

SUM(bs.Runs\_Scored) AS Total\_Score

FROM ipl\_Batsman\_Scored bs

JOIN ipl\_match m ON bs.Match\_Id = m.Match\_Id

GROUP BY m.Season\_Id,

CASE

WHEN bs.Innings\_No = 1 THEN m.Team\_1

WHEN bs.Innings\_No = 2 THEN m.Team\_2

END

)

SELECT t.Team\_Name, ts.Season\_Id, AVG(ts.Total\_Score) AS Average\_Score

FROM Team\_Scores ts

JOIN ipl\_Team t ON ts.Team\_Id = t.Team\_Id

GROUP BY t.Team\_Name, ts.Season\_Id

ORDER BY t.Team\_Name, ts.Season\_Id;

This query calculates the average score of each team per season.

**7.Most Common Outcomes of Matches**

CREATE TABLE Outcome AS

SELECT o.Outcome\_Type, COUNT(m.Match\_Id) AS Outcome\_Count

FROM ipl\_match m

JOIN ipl\_Outcome o ON m.Outcome\_type = o.Outcome\_Id

GROUP BY o.Outcome\_Type

ORDER BY Outcome\_Count DESC;

This query lists the most common outcomes of matches.

**8.Performance of Teams by Venue**

CREATE TABLE match\_win\_venue AS

SELECT v.Venue\_Name, t.Team\_Name, COUNT(m.Match\_Id) AS Matches\_Won

FROM ipl\_match m

JOIN ipl\_Team t ON m.Match\_Winner = t.Team\_Id

JOIN ipl\_Venue v ON m.Venue\_Id = v.Venue\_Id

GROUP BY v.Venue\_Name, t.Team\_Name

ORDER BY v.Venue\_Name, Matches\_Won DESC;

This query analyses the performance of teams by venue.

**9.Average Number of Runs Scored per Match**

CREATE TABLE Average\_Match\_runs AS

WITH Match\_Runs AS (

SELECT m.Match\_Id, SUM(bs.Runs\_Scored) AS Total\_Runs

FROM ipl\_Batsman\_Scored bs

JOIN ipl\_match m ON bs.Match\_Id = m.Match\_Id

GROUP BY m.Match\_Id

)

SELECT AVG(mr.Total\_Runs) AS Average\_Runs\_Per\_Match

FROM Match\_Runs mr;

This query calculates the average number of runs scored per match.

**10.Players with the Highest Number of Sixes**

CREATE TABLE sixes AS

SELECT \* FROM (

SELECT pm.match\_id, player\_name, COUNT(runs\_scored) AS total\_six

FROM ipl\_batsman\_scored bs

JOIN ipl\_player\_match pm ON pm.match\_id = bs.match\_id

JOIN ipl\_player p ON p.Player\_id = pm.Player\_id

WHERE Runs\_scored = 6

GROUP BY pm.match\_id, player\_name

) AS A

ORDER BY total\_six DESC;

This query creates a table listing players with the highest number of sixes.

### **Conclusion**

The above SQL queries provide a comprehensive analysis of the IPL dataset, offering insights into match statistics, player performances, team strategies, and more. This project demonstrates the power of SQL in extracting and analysing data from a relational database.